In [2]:
```
pip install web3
```

```
Requirement already satisfied: web3 in ./anaconda3/lib/python3.11/site-pa
ckages (6.15.1)
Requirement already satisfied: aiohttp>=3.7.4.post0 in ./anaconda3/lib/py
thon3.11/site-packages (from web3) (3.8.5)
Requirement already satisfied: eth-abi>=4.0.0 in ./anaconda3/lib/python3.
11/site-packages (from web3) (5.0.0)
Requirement already satisfied: eth-account>=0.8.0 in ./anaconda3/lib/pyth
on3.11/site-packages (from web3) (0.11.0)
Requirement already satisfied: eth-hash[pycryptodome]>=0.5.1 in ./anacond
a3/lib/python3.11/site-packages (from web3) (0.6.0)
Requirement already satisfied: eth-typing>=3.0.0 in ./anaconda3/lib/pytho
n3.11/site-packages (from web3) (3.5.2)
Requirement already satisfied: eth-utils>=2.1.0 in ./anaconda3/lib/python
3.11/site-packages (from web3) (2.3.1)
Requirement already satisfied: hexbytes<0.4.0,>=0.1.0 in ./anaconda3/lib/
python3.11/site-packages (from web3) (0.3.1)
Requirement already satisfied: jsonschema>=4.0.0 in ./anaconda3/lib/pytho
n3.11/site-packages (from web3) (4.17.3)
Requirement already satisfied: lru-dict<1.3.0,>=1.1.6 in ./anaconda3/lib/
python3.11/site-packages (from web3) (1.2.0)
Requirement already satisfied: protobuf>=4.21.6 in ./anaconda3/lib/python
3.11/site-packages (from web3) (4.25.2)
Requirement already satisfied: requests>=2.16.0 in ./anaconda3/lib/python
3.11/site-packages (from web3) (2.31.0)
Requirement already satisfied: typing-extensions>=4.0.1 in ./anaconda3/li
b/python3.11/site-packages (from web3) (4.7.1)
Requirement already satisfied: websockets>=10.0.0 in ./anaconda3/lib/pyth
on3.11/site-packages (from web3) (12.0)
Requirement already satisfied: pyunormalize>=15.0.0 in ./anaconda3/lib/py
thon3.11/site-packages (from web3) (15.1.0)
Requirement already satisfied: attrs>=17.3.0 in ./anaconda3/lib/python3.1
1/site-packages (from aiohttp>=3.7.4.post0->web3) (22.1.0)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in ./anaconda
3/lib/python3.11/site-packages (from aiohttp>=3.7.4.post0->web3) (2.0.4)
Requirement already satisfied: multidict<7.0,>=4.5 in ./anaconda3/lib/pyt
hon3.11/site-packages (from aiohttp>=3.7.4.post0->web3) (6.0.2)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in ./anaconda
3/lib/python3.11/site-packages (from aiohttp>=3.7.4.post0->web3) (4.0.2)
Requirement already satisfied: yarl<2.0,>=1.0 in ./anaconda3/lib/python3.
11/site-packages (from aiohttp>=3.7.4.post0->web3) (1.8.1)
Requirement already satisfied: frozenlist>=1.1.1 in ./anaconda3/lib/pytho
n3.11/site-packages (from aiohttp>=3.7.4.post0->web3) (1.3.3)
Requirement already satisfied: aiosignal>=1.1.2 in ./anaconda3/lib/python
3.11/site-packages (from aiohttp>=3.7.4.post0->web3) (1.2.0)
Requirement already satisfied: parsimonious<0.10.0,>=0.9.0 in ./anaconda
3/lib/python3.11/site-packages (from eth-abi>=4.0.0->web3) (0.9.0)
Requirement already satisfied: bitarray>=2.4.0 in ./anaconda3/lib/python
3.11/site-packages (from eth-account>=0.8.0->web3) (2.9.2)
Requirement already satisfied: eth-keyfile>=0.6.0 in ./anaconda3/lib/pyth
on3.11/site-packages (from eth-account>=0.8.0->web3) (0.7.0)
Requirement already satisfied: eth-keys>=0.4.0 in ./anaconda3/lib/python
3.11/site-packages (from eth-account>=0.8.0->web3) (0.4.0)
Requirement already satisfied: eth-rlp>=0.3.0 in ./anaconda3/lib/python3.
```

```
11/site-packages (from eth-account>=0.8.0->web3) (1.0.1)
Requirement already satisfied: rlp>=1.0.0 in ./anaconda3/lib/python3.11/s
ite-packages (from eth-account>=0.8.0->web3) (3.0.0)
Requirement already satisfied: pycryptodome<4,>=3.6.6 in ./anaconda3/lib/
python3.11/site-packages (from eth-hash[pycryptodome]>=0.5.1->web3) (3.2
0.0)
Requirement already satisfied: cytoolz>=0.10.1 in ./anaconda3/lib/python
3.11/site-packages (from eth-utils>=2.1.0->web3) (0.12.0)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.1
4.0 in ./anaconda3/lib/python3.11/site-packages (from jsonschema>=4.0.0->
web3) (0.18.0)
Requirement already satisfied: idna<4,>=2.5 in ./anaconda3/lib/python3.1
1/site-packages (from requests>=2.16.0->web3) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in ./anaconda3/lib/pyth
on3.11/site-packages (from requests>=2.16.0->web3) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in ./anaconda3/lib/pyth
on3.11/site-packages (from requests>=2.16.0->web3) (2023.7.22)
Requirement already satisfied: toolz>=0.8.0 in ./anaconda3/lib/python3.1
1/site-packages (from cytoolz>=0.10.1->eth-utils>=2.1.0->web3) (0.12.0)
Requirement already satisfied: regex>=2022.3.15 in ./anaconda3/lib/python
3.11/site-packages (from parsimonious<0.10.0,>=0.9.0->eth-abi>=4.0.0->web
3) (2022.7.9)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:

```
pip install 'web3[tester]'
```

```
Requirement already satisfied: web3[tester] in ./anaconda3/lib/python3.1
1/site-packages (6.15.1)
Requirement already satisfied: aiohttp>=3.7.4.post0 in ./anaconda3/lib/py
thon3.11/site-packages (from web3[tester]) (3.8.5)
Requirement already satisfied: eth-abi>=4.0.0 in ./anaconda3/lib/python3.
11/site-packages (from web3[tester]) (5.0.0)
Requirement already satisfied: eth-account>=0.8.0 in ./anaconda3/lib/pyth
on3.11/site-packages (from web3[tester]) (0.11.0)
Requirement already satisfied: eth-hash[pycryptodome]>=0.5.1 in ./anacond
a3/lib/python3.11/site-packages (from web3[tester]) (0.6.0)
Requirement already satisfied: eth-typing>=3.0.0 in ./anaconda3/lib/pytho
n3.11/site-packages (from web3[tester]) (3.5.2)
Requirement already satisfied: eth-utils>=2.1.0 in ./anaconda3/lib/python
3.11/site-packages (from web3[tester]) (2.3.1)
Requirement already satisfied: hexbytes<0.4.0,>=0.1.0 in ./anaconda3/lib/
python3.11/site-packages (from web3[tester]) (0.3.1)
Requirement already satisfied: jsonschema>=4.0.0 in ./anaconda3/lib/pytho
n3.11/site-packages (from web3[tester]) (4.17.3)
Requirement already satisfied: lru-dict<1.3.0,>=1.1.6 in ./anaconda3/lib/
python3.11/site-packages (from web3[tester]) (1.2.0)
Requirement already satisfied: protobuf>=4.21.6 in ./anaconda3/lib/python
3.11/site-packages (from web3[tester]) (4.25.2)
Requirement already satisfied: requests>=2.16.0 in ./anaconda3/lib/python
3.11/site-packages (from web3[tester]) (2.31.0)
Requirement already satisfied: typing-extensions>=4.0.1 in ./anaconda3/li
b/python3.11/site-packages (from web3[tester]) (4.7.1)
Requirement already satisfied: websockets>=10.0.0 in ./anaconda3/lib/pyth
on3.11/site-packages (from web3[tester]) (12.0)
Requirement already satisfied: pyunormalize>=15.0.0 in ./anaconda3/lib/py
thon3.11/site-packages (from web3[tester]) (15.1.0)
Requirement already satisfied: eth-tester[py-evm]==v0.9.1-b.2 in ./anacon
```

```
da3/lib/python3.11/site-packages (from web3[tester]) (0.9.1b2)
Requirement already satisfied: py-geth>=3.14.0 in ./anaconda3/lib/python
3.11/site-packages (from web3[tester]) (4.2.0)
Requirement already satisfied: eth-keys<0.5.0,>=0.4.0 in ./anaconda3/lib/
python3.11/site-packages (from eth-tester[py-evm]==v0.9.1-b.2->web3[teste
r]) (0.4.0)
Requirement already satisfied: rlp<4,>=3.0.0 in ./anaconda3/lib/python3.1
1/site-packages (from eth-tester[py-evm]==v0.9.1-b.2->web3[tester]) (3.0.
0)
Requirement already satisfied: semantic-version<3.0.0,>=2.6.0 in ./anacon
da3/lib/python3.11/site-packages (from eth-tester[py-evm]==v0.9.1-b.2->we
b3[tester]) (2.10.0)
Requirement already satisfied: py-evm==0.7.0a4 in ./anaconda3/lib/python
3.11/site-packages (from eth-tester[py-evm]==v0.9.1-b.2->web3[tester]) (
0.7.0a4)
Requirement already satisfied: cached-property<2,>=1.5.1 in ./anaconda3/l
ib/python3.11/site-packages (from py-evm==0.7.0a4->eth-tester[py-evm]==v
0.9.1-b.2->web3[tester]) (1.5.2)
Requirement already satisfied: eth-bloom>=1.0.3 in ./anaconda3/lib/python
3.11/site-packages (from py-evm==0.7.0a4->eth-tester[py-evm]==v0.9.1-b.2-
>web3[tester]) (3.0.0)
Requirement already satisfied: mypy-extensions>=1.0.0 in ./anaconda3/lib/
python3.11/site-packages (from py-evm==0.7.0a4->eth-tester[py-evm]==v0.9.
1-b.2->web3[tester]) (1.0.0)
Requirement already satisfied: py-ecc<7.0.0,>=1.4.7 in ./anaconda3/lib/py
thon3.11/site-packages (from py-evm==0.7.0a4->eth-tester[py-evm]==v0.9.1-
b.2->web3[tester]) (6.0.0)
Requirement already satisfied: pyethash<1.0.0,>=0.1.27 in ./anaconda3/li
b/python3.11/site-packages (from py-evm==0.7.0a4->eth-tester[py-evm]==v0.
9.1-b.2->web3[tester]) (0.1.27)
Requirement already satisfied: trie<3,>=2.0.0 in ./anaconda3/lib/python3.
11/site-packages (from py-evm==0.7.0a4->eth-tester[py-evm]==v0.9.1-b.2->w
eb3[tester]) (2.2.0)
Requirement already satisfied: attrs>=17.3.0 in ./anaconda3/lib/python3.1
1/site-packages (from aiohttp>=3.7.4.post0->web3[tester]) (22.1.0)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in ./anaconda
3/lib/python3.11/site-packages (from aiohttp>=3.7.4.post0->web3[tester])
(2.0.4)
Requirement already satisfied: multidict<7.0,>=4.5 in ./anaconda3/lib/pyt
hon3.11/site-packages (from aiohttp>=3.7.4.post0->web3[tester]) (6.0.2)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in ./anaconda
3/lib/python3.11/site-packages (from aiohttp>=3.7.4.post0->web3[tester])
(4.0.2)
Requirement already satisfied: yarl<2.0,>=1.0 in ./anaconda3/lib/python3.
11/site-packages (from aiohttp>=3.7.4.post0->web3[tester]) (1.8.1)
Requirement already satisfied: frozenlist>=1.1.1 in ./anaconda3/lib/pytho
n3.11/site-packages (from aiohttp>=3.7.4.post0->web3[tester]) (1.3.3)
Requirement already satisfied: aiosignal>=1.1.2 in ./anaconda3/lib/python
3.11/site-packages (from aiohttp>=3.7.4.post0->web3[tester]) (1.2.0)
Requirement already satisfied: parsimonious<0.10.0,>=0.9.0 in ./anaconda
3/lib/python3.11/site-packages (from eth-abi>=4.0.0->web3[tester]) (0.9.
0)
Requirement already satisfied: bitarray>=2.4.0 in ./anaconda3/lib/python
3.11/site-packages (from eth-account>=0.8.0->web3[tester]) (2.9.2)
Requirement already satisfied: eth-keyfile>=0.6.0 in ./anaconda3/lib/pyth
on3.11/site-packages (from eth-account>=0.8.0->web3[tester]) (0.7.0)
Requirement already satisfied: eth-rlp>=0.3.0 in ./anaconda3/lib/python3.
```

```
11/site-packages (from eth-account>=0.8.0->web3[tester]) (1.0.1)
Requirement already satisfied: pycryptodome<4,>=3.6.6 in ./anaconda3/lib/
python3.11/site-packages (from eth-hash[pycryptodome]>=0.5.1->web3[teste
r]) (3.20.0)
Requirement already satisfied: cytoolz>=0.10.1 in ./anaconda3/lib/python
3.11/site-packages (from eth-utils>=2.1.0->web3[tester]) (0.12.0)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.1
4.0 in ./anaconda3/lib/python3.11/site-packages (from jsonschema>=4.0.0->
web3[tester]) (0.18.0)
Requirement already satisfied: idna<4,>=2.5 in ./anaconda3/lib/python3.1
1/site-packages (from requests>=2.16.0->web3[tester]) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in ./anaconda3/lib/pyth
on3.11/site-packages (from requests>=2.16.0->web3[tester]) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in ./anaconda3/lib/pyth
on3.11/site-packages (from requests>=2.16.0->web3[tester]) (2023.7.22)
Requirement already satisfied: toolz>=0.8.0 in ./anaconda3/lib/python3.1
1/site-packages (from cytoolz>=0.10.1->eth-utils>=2.1.0->web3[tester]) (
0.12.0)
Requirement already satisfied: safe-pysha3>=1.0.0 in ./anaconda3/lib/pyth
on3.11/site-packages (from eth-hash[pycryptodome]>=0.5.1->web3[tester]) (
1.0.4)
Requirement already satisfied: regex>=2022.3.15 in ./anaconda3/lib/python
3.11/site-packages (from parsimonious<0.10.0,>=0.9.0->eth-abi>=4.0.0->web
3[tester]) (2022.7.9)
Requirement already satisfied: sortedcontainers>=2.1.0 in ./anaconda3/li
b/python3.11/site-packages (from trie<3,>=2.0.0->py-evm==0.7.0a4->eth-tes
ter[py-evm]==v0.9.1-b.2->web3[tester]) (2.4.0)
Note: you may need to restart the kernel to use updated packages.
```

In [4]:

```python
from web3 import Web3
```

In [5]:

```python
Web3.to_wei(1,'ether')
```

Out[5]:

```
1000000000000000000
```

In [6]:

```python
Web3.from_wei(5000000000,'gwei')
```

Out[6]:

```
Decimal('5')
```

In [7]:

```python
Web3.from_wei(10000000000000000000, 'ether')
```

Out[7]:

```
Decimal('10')
```

In [8]:

```python
## w3 = Web3(Web3.HTTPProvider('http://127.0.0.1:8545')) to configure to
```

In [9]:

```python
#connecting to simulated ethereum node through ethereumtestprovider which
w3 = Web3(Web3.EthereumTesterProvider())
```

In [10]:

```python
w3.is_connected()
```

Out[10]:

```
True
```

In [11]:

```python
w3.eth.accounts  ## checking the public key of test eth accounts
```

```
Out[11]: ['0x7E5F4552091A69125d5DfCb7b8C2659029395Bdf',
          '0x2B5AD5c4795c026514f8317c7a215E218DcCD6cF',
          '0x6813Eb9362372EEF6200f3b1dbC3f819671cBA69',
          '0x1efF47bc3a10a45D4B230B5d10E37751FE6AA718',
          '0xe1AB8145F7E55DC933d51a18c793F901A3A0b276',
          '0xE57bFE9F44b819898F47BF37E5AF72a0783e1141',
          '0xd41c057fd1c78805AAC12B0A94a405c0461A6FBb',
          '0xF1F6619B38A98d6De0800F1DefC0a6399eB6d30C',
          '0xF7Edc8FA1eCc32967F827C9043FcAe6ba73afA5c',
          '0x4CCeBa2d7D2B4fdcE4304d3e09a1fea9fbEb1528']
```

```
In [12]: w3.eth.get_balance(w3.eth.accounts[0])  #ether values are represented in
```

```
Out[12]: 1000000000000000000000000
```

```
In [13]: w3.from_wei(1000000000000000000000000, 'ether')  #Converting wei to ether
```

```
Out[13]: Decimal('1000000')
```

```
In [14]: w3.eth.get_block('latest')
         #A lot of information gets returned about a block, but just a couple thin

         #The block number is zero — no matter how long ago you configured the tes
         #transactions is an empty list, for the same reason: we haven’t done anyt
         #Notice that the parentHash is just a bunch of empty bytes. This signifie
```

Out[14]:

```
AttributeDict({'number': 0,
 'hash': HexBytes('0x5cd4adc3481c632fea8f05ec2338c2c9e78384279d50efd07cfa
7df9cf70b418'),
 'parentHash': HexBytes('0x0000000000000000000000000000000000000000000000
000000000000000000'),
 'nonce': HexBytes('0x0000000000000000'),
 'sha3Uncles': HexBytes('0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a74
13f0a142fd40d49347'),
 'logsBloom': HexBytes('0x00'),
 'transactionsRoot': HexBytes('0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b
996cadc001622fb5e363b421'),
 'receiptsRoot': HexBytes('0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996c
adc001622fb5e363b421'),
 'stateRoot': HexBytes('0xf1588db9a9f1ed91effabdec31f93cb4212b008c8b8ba04
7fd55fabebf6fd727'),
 'miner': '0x0000000000000000000000000000000000000000',
 'difficulty': 0,
 'totalDifficulty': 0,
 'mixHash': HexBytes('0x0000000000000000000000000000000000000000000000000
000000000000000'),
 'size': 548,
 'extraData': HexBytes('0x00000000000000000000000000000000000000000000000
00000000000000000'),
 'gasLimit': 30029122,
 'gasUsed': 0,
 'timestamp': 1707469870,
 'transactions': [],
 'uncles': [],
 'baseFeePerGas': 1000000000,
 'withdrawals': [],
 'withdrawalsRoot': HexBytes('0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b9
96cadc001622fb5e363b421')})
```

In [15]:

```
#sending ether from account 1 to account3
tx_hash=w3.eth.send_transaction({'from':w3.eth.accounts[0], 'to':w3.eth.a
```

In [16]:

```
w3.eth.wait_for_transaction_receipt(tx_hash)
```

Out[16]:

```
AttributeDict({'blockHash': HexBytes('0x0c237f1ab540d4a2ddbafede84063eecc
dcd4a125160a351537c2e88f469bdf9'),
 'blockNumber': 1,
 'contractAddress': None,
 'cumulativeGasUsed': 21000,
 'effectiveGasPrice': 1000000000,
 'from': '0x7E5F4552091A69125d5DfCb7b8C2659029395Bdf',
 'gasUsed': 21000,
 'logs': [],
 'state_root': b'\x01',
 'status': 1,
 'to': '0x6813Eb9362372EEF6200f3b1dbC3f819671cBA69',
 'transactionHash': HexBytes('0xbb774d5866e392c98db85afe4ce50c53032d081b5
5cd3e7d141e2d5a3a9bc09c'),
 'transactionIndex': 0,
 'type': 2})
```

In [17]:

```
w3.eth.get_transaction(tx_hash)
```

Out[17]:
```
AttributeDict({'type': 2,
 'hash': HexBytes('0xbb774d5866e392c98db85afe4ce50c53032d081b55cd3e7d141e
2d5a3a9bc09c'),
 'nonce': 0,
 'blockHash': HexBytes('0x0c237f1ab540d4a2ddbafede84063eeccdcd4a125160a35
1537c2e88f469bdf9'),
 'blockNumber': 1,
 'transactionIndex': 0,
 'from': '0x7E5F4552091A69125d5DfCb7b8C2659029395Bdf',
 'to': '0x6813Eb9362372EEF6200f3b1dbC3f819671cBA69',
 'value': 5000000000000000000,
 'gas': 21000,
 'data': HexBytes('0x'),
 'r': HexBytes('0x062f15f05c8b957d68b3a0704adbbd71c2fd5c92fa01407565adc5d
99c799d03'),
 's': HexBytes('0x7986c95357643f523cfc4a0bbf0ef5e07bcfe22ddfa37a60c1fc3b0
18e650692'),
 'v': 0,
 'chainId': 131277322940537,
 'maxFeePerGas': 1000000000,
 'maxPriorityFeePerGas': 1000000000,
 'accessList': [],
 'gasPrice': 1000000000})
```

In [18]:
```
#verifying account balance from account 1 and account3
w3.eth.get_balance(w3.eth.accounts[0])
#the higher difference in the remaining wei is because of the gas burned
```

Out[18]:
```
999994999979000000000000
```

In [19]:
```
w3.eth.get_balance(w3.eth.accounts[2])
```

Out[19]:
```
1000005000000000000000000
```

In [20]:
```
acct1 = w3.eth.accounts[0]
```

In [21]:
```
tx=w3.eth.get_transaction(tx_hash)
assert tx['from']== acct1 ## Assert that the "from" address of the transa
```

In [22]:
```
from web3.middleware import construct_sign_and_send_raw_middleware

import os
from dotenv import load_dotenv
load_dotenv()
```

Out[22]:
```
False
```

In [23]:
```
# Example private key:
pk = 0x7ec01898a720beda9304921aec62735846aa1a545a8332bc030577c21f72cd46
```

In [24]:
```
# Note: Never commit your key in your code! Use env variables instead:
#syntax: pk=os.environ.get("PRIVATE_KEY")
```

In [25]:
```
#creating a representation of an Ethereum account using a private key.
acct2 = w3.eth.account.from_key(pk)
```

In [26]:
```
##sending new transaction
w3.eth.send_transaction({'from':acct1, 'value':w3.to_wei(3, 'ether'), 'to
```

Out[26]:
```
HexBytes('0xab907df3248bfcf1601be4effee5b17d8e343cdcc6bad6d7ddb3ea921e78e
9ed')
```

In [27]:
```
print(f"acct2 balance: {w3.eth.get_balance(acct2.address)}")
```

```
acct2 balance: 3000000000000000000
```

In [28]:
```
##Adding acct2 as autosigner
w3.middleware_onion.add(construct_sign_and_send_raw_middleware(acct2))
```

In [29]:
```
tx_hash = w3.eth.send_transaction({'from':acct2.address, 'value':3, 'to':
tx= w3.eth.get_transaction(tx_hash)
```

In [30]:
```
assert tx['from']== acct2.address
```

In [31]:
```
w3.eth.get_balance(acct2.address,'latest')
```

Out[31]:
```
2999964926018147997
```

In [32]:
```
##optionally we can set a default signer as well
##syntax for setting a default signer is as follows: w3.eth.default_accou
## Then, if we omit a "from" key, acct2 will be used.
```

In [33]:
```
#if we don't opt for the middleware, we will need to:

#build each transaction,
#sign_transaction, and
#then use send_raw_transaction.
```

In [34]:
```
#building a new transaction
transaction ={'from':acct2.address, 'value':3, 'to':w3.eth.accounts[3],
               'nonce':w3.eth.get_transaction_count(acct2.address),
              'gas':200000,'maxFeePerGas':2000000000,'maxPriorityFeePerGas
```

In [35]:
```
#signing transaction with a private key
signed= w3.eth.account.sign_transaction(transaction, pk)
```

In [36]:
```
#the raw transaction is what will be broadcasted to the network
print(f" raw tx: {signed.rawTransaction.hex()}")
```

```
 raw tx: 0x02f86b8001843b9aca008477359400830 30d40941eff47bc3a10a45d4b230b
5d10e37751fe6aa7180380c001a0ca53cf61a11b8f9a50b0b53708133527341fb2c2fcb00
f23fff3783513141a49a055b49b62200b79073e4f69d72cfea8a31c8eeceb74db06a86984
8e6c7ea5e050
```

In [37]:

```
##now sending the signed transaction
tx_hash= w3.eth.send_raw_transaction(signed.rawTransaction)
tx= w3.eth.get_transaction(tx_hash)
assert tx['from']== acct2.address #verifying the whether the transaction
```

In [43]:

```
#Python code that deploy and interact with a simple Ethereum smart contra
# // SPDX-License-Identifier: MIT
# pragma solidity 0.8.17;
#
# contract Billboard {
#     string public message;
#
#     constructor(string memory _message) {
#         message = _message;
#     }
#
#     function writeBillboard(string memory _message) public {
#         message = _message;
#     }
# }


#Use a Solidity compiler like solc to compile the code.
#For example, if you have a Solidity file named Billboard.sol, you can co
#solc --bin --abi --optimize -o <output_directory> Billboard.sol(This com
#The ABI is a JSON-formatted representation of the smart contract's inter
#the ABI is crucial for interacting with the smart contract, while the by

init_bytecode = "608060405234801562000011576000080fd5b5060405162000bee3803
abi = '[{"inputs": [{"internalType": "string","name": "_message","type":

# Initializing the contract factory:
Billboard= w3.eth.contract(bytecode=init_bytecode, abi=abi)

#deploying a contract using `transact` + the signer middleware:
tx_hash = Billboard.constructor("Gm").transact({"from": acct2.address})

receipt=w3.eth.get_transaction_receipt(tx_hash)
deployed_address= receipt["contractAddress"]
print(f"Billboard contract deployed at address: {deployed_address}")
billboard = w3.eth.contract(address=deployed_address, abi=abi)
billboard.all_functions()
```

```
Billboard contract deployed at address: 0xB50219400530edE87dBdFDd64D8b00b
97E0c614d
```

Out[43]:

```
[<Function message()>, <Function writeBillboard(string)>]
```

In [48]:

```
from pprint import pprint
#Manually build and sign a transaction
unsent_billboard_tx = billboard.functions.writeBillboard("Gn").build_tran
    "from": acct2.address,
    "nonce": w3.eth.get_transaction_count(acct2.address),})
pprint(unsent_billboard_tx)
```

```
{'chainId': 131277322940537,
 'data': '0x6153621100000000000000000000000000000000000000000000000000000
0000000002000000000000000000000000000000000000000000000000000000000000000
02476e000000000000000000000000000000000000000000000000000000000000',
 'from': '0x043414E7Ec0b1dDdD7d2a69b53c2bA498F3167cB',
 'gas': 36134,
 'maxFeePerGas': 1696707400,
 'maxPriorityFeePerGas': 1000000000,
 'nonce': 6,
 'to': '0xB50219400530edE87dBdFDd64D8b00b97E0c614d',
 'value': 0}
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```